**Go static code analysis**

Unique rules to find Bugs, Security Hotspots, and Code Smells in your GO code

### [Hard-coded credentials are security-sensitive](https://rules.sonarsource.com/go/RSPEC-2068)

[Security Hotspot](https://rules.sonarsource.com/go/RSPEC-2068)

### [Cognitive Complexity of functions should not be too high](https://rules.sonarsource.com/go/RSPEC-3776)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-3776)

### [String literals should not be duplicated](https://rules.sonarsource.com/go/RSPEC-1192)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1192)

### [Functions should not be empty](https://rules.sonarsource.com/go/RSPEC-1186)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1186)

### [All branches in a conditional structure should not have exactly the same implementation](https://rules.sonarsource.com/go/RSPEC-3923)

[Bug](https://rules.sonarsource.com/go/RSPEC-3923)

### ["=+" should not be used instead of "+="](https://rules.sonarsource.com/go/RSPEC-2757)

[Bug](https://rules.sonarsource.com/go/RSPEC-2757)

### [Related "if/else if" statements should not have the same condition](https://rules.sonarsource.com/go/RSPEC-1862)

[Bug](https://rules.sonarsource.com/go/RSPEC-1862)

### [Identical expressions should not be used on both sides of a binary operator](https://rules.sonarsource.com/go/RSPEC-1764)

[Bug](https://rules.sonarsource.com/go/RSPEC-1764)

### [All code should be reachable](https://rules.sonarsource.com/go/RSPEC-1763)

[Bug](https://rules.sonarsource.com/go/RSPEC-1763)

### [Variables should not be self-assigned](https://rules.sonarsource.com/go/RSPEC-1656)

[Bug](https://rules.sonarsource.com/go/RSPEC-1656)

### [Functions should not have identical implementations](https://rules.sonarsource.com/go/RSPEC-4144)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-4144)

### [Two branches in a conditional structure should not have exactly the same implementation](https://rules.sonarsource.com/go/RSPEC-1871)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1871)

### ["switch" statements should not have too many "case" clauses](https://rules.sonarsource.com/go/RSPEC-1479)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1479)

### [Track uses of "FIXME" tags](https://rules.sonarsource.com/go/RSPEC-1134)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1134)

### [Redundant pairs of parentheses should be removed](https://rules.sonarsource.com/go/RSPEC-1110)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1110)

### [Nested blocks of code should not be left empty](https://rules.sonarsource.com/go/RSPEC-108)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-108)

### [Functions should not have too many parameters](https://rules.sonarsource.com/go/RSPEC-107)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-107)

### [Using hardcoded IP addresses is security-sensitive](https://rules.sonarsource.com/go/RSPEC-1313)

[Security Hotspot](https://rules.sonarsource.com/go/RSPEC-1313)

### [Multi-line comments should not be empty](https://rules.sonarsource.com/go/RSPEC-4663)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-4663)

### [Boolean checks should not be inverted](https://rules.sonarsource.com/go/RSPEC-1940)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1940)

### [Local variable and function parameter names should comply with a naming convention](https://rules.sonarsource.com/go/RSPEC-117)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-117)

### [Boolean literals should not be redundant](https://rules.sonarsource.com/go/RSPEC-1125)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1125)

### [Function and method names should comply with a naming convention](https://rules.sonarsource.com/go/RSPEC-100)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-100)

### [Track uses of "TODO" tags](https://rules.sonarsource.com/go/RSPEC-1135)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1135)

### [Track lack of copyright and license headers](https://rules.sonarsource.com/go/RSPEC-1451)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1451)

### [Octal values should not be used](https://rules.sonarsource.com/go/RSPEC-1314)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1314)

### ["switch" statements should not be nested](https://rules.sonarsource.com/go/RSPEC-1821)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1821)

### [Control flow statements "if", "for" and "switch" should not be nested too deeply](https://rules.sonarsource.com/go/RSPEC-134)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-134)

### ["switch" statements should have "default" clauses](https://rules.sonarsource.com/go/RSPEC-131)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-131)

### ["if ... else if" constructs should end with "else" clauses](https://rules.sonarsource.com/go/RSPEC-126)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-126)

### [Expressions should not be too complex](https://rules.sonarsource.com/go/RSPEC-1067)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1067)

### [Useless "if(true) {...}" and "if(false){...}" blocks should be removed](https://rules.sonarsource.com/go/RSPEC-1145)

[Bug](https://rules.sonarsource.com/go/RSPEC-1145)

### [Track parsing failures](https://rules.sonarsource.com/go/RSPEC-2260)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-2260)

### [Functions and methods should not have too many lines](https://rules.sonarsource.com/go/RSPEC-138)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-138)

### [Statements should be on separate lines](https://rules.sonarsource.com/go/RSPEC-122)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-122)

### ["switch case" clauses should not have too many lines](https://rules.sonarsource.com/go/RSPEC-1151)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-1151)

### [Files should not have too many lines of code](https://rules.sonarsource.com/go/RSPEC-104)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-104)

### [Lines should not be too long](https://rules.sonarsource.com/go/RSPEC-103)

[Code Smell](https://rules.sonarsource.com/go/RSPEC-103)